AMENDMENTS TO THE CLAIMS:

Complete Listing of Claims

Claim 1. (original) An encoder for generating a code word according to a low-density parity check code, comprising:

a matrix value memory for storing shift and weight parameters associated with each of a plurality of macro matrix entries;

a register for storing at least a portion of an information bit vector; and

a cyclic multiply unit, coupled to the matrix value memory and to the register, and comprising:

a shifter for circularly shifting a selected portion of the information bit vector;

a first accumulator, for storing a result value; and

an exclusive-OR function, coupled to the shifter and to the accumulator, for performing a bitwise exclusive-OR operation on the output of the shifter and the output of the first accumulator; and

a right-hand side value memory, coupled to the cyclic multiply unit, for storing product values.

Claim 2. (original) The encoder of claim 1, further comprising:

a parity memory, coupled to the cyclic multiply unit, for storing parity bits for the code word.

Claim 3. (original) The encoder of claim 1, wherein the cyclic multiply unit further comprises:

a second accumulator, having a first input coupled to the right-hand side value memory and a second input coupled to the output of the first accumulator, for updating the contents of the right-hand side value with the contents of the first accumulator.

Claim 4. (original) The encoder of claim 1, further comprising:

a controller, for controlling the encoder responsive to a sequence of program instructions.

Claim 5. (original) The encoder of claim 1, further comprising:

a parity memory, coupled to the cyclic multiply unit, for storing parity bits for the code word; and

an input multiplexer, having inputs coupled to the register, to the right-hand side value memory, and to the parity memory, for applying a selected value to the shifter of the cyclic multiply unit.

Claim 6. (original) The encoder of claim 1, further comprising:

an inverse submatrix memory, for storing parameters corresponding to an inverse of a submatrix; and

an inverse multiply unit, for multiplying product values by parameters stored in the inverse submatrix memory.

Claim 7. (original) The encoder of claim 6, wherein the inverse submatrix memory is for storing a sequence of parameters for entries of the inverse of the submatrix;

and wherein the inverse multiply unit comprises:

a first register for storing a first portion of a product value;

a second register for storing a second portion of the product value;

a first logic circuit for combining the sequence of parameters with the first portion of the product value; and

a second logic circuit for combining the sequence of parameters, in a transposed sequence relative to that combined by the first logic circuit, with the second portion of the product value.

Claim 8. (original) The encoder of claim 7, wherein the inverse multiply unit further comprises:

a result accumulator for accumulating results generated by the first and second logic circuits.

Claim 9. (original) The encoder of claim 7, wherein the parameters stored in the inverse submatrix memory correspond to parameters along the diagonal and on one side of the diagonal of the submatrix.

Claim 10. (original) The encoder of claim 1, further comprising:

a parity memory, coupled to the cyclic multiply unit, for storing parity bits for the code word; and

a weight-two solution unit, having an input for receiving values from the right-hand side value memory, and having an output coupled to the parity memory, for successively solving parity bit values in a submatrix having a weight of two.

Claim 11. (original) The encoder of claim 10, wherein the weight-two solution unit comprises:

a register for storing values from the right-hand side value memory;

a multiplexer, coupled to the register, for selecting one of the bits of the register responsive to a control input;

a row index register, for storing a shifted value and for applying the value to the multiplexer as its control input;

logic circuitry, for generating an exclusive-OR of a prior result value and the bit selected by the multiplexer;

an output register, for storing results from the logic circuitry;

a demultiplexer, for coupling the logic circuitry to a position of the output register responsive to a control input; and

a column index register, for storing a shifted value and for applying the value to the demultiplexer as its control input.

Claim 12. (original) A method of encoding code words according to a low-density parity check matrix, comprising the steps of:

storing, in a memory, parameters for each of a plurality of non-zero entries of a parity check matrix, the parameters for each entry comprising a weight value indicating the number of shifted diagonals contained within that entry, and shift values indicating the position of the shifted diagonals;

receiving an information word;

multiplying the information word by a portion of the parameters of the parity check matrix;

storing the results of the multiplying step in a right-hand side value memory;

for one or more submatrices, successively solving a plurality of parity bits using results from the right-hand side value memory and stored parameters of the submatrix in the memory;

then updating the contents of the right-hand side value memory for columns corresponding to the plurality of parity bits;

for a remaining identity portion of the parity check matrix, determining corresponding parity bits from the stored right-hand side values.

Claim 13. (original) The method of claim 12, wherein the successively solving step and the updating step are repeated for a plurality of submatrices.

Claim 14. (original) The method of claim 12, further comprising:

prior to the step of storing parameters for each of a plurality of non-zero entries of a parity check matrix, selecting a low-density parity check code;

factoring a parity check matrix corresponding to the selected code so that a left-hand portion comprises an identity portion above a zero portion, and at least one submatrix in a lower-right portion of the left-hand portion;

storing weight values and shift values for each of the entries of the rearranged parity check matrix.

Claim 15. (original) The method of claim 12, wherein the at least one submatrix comprises a diagonal submatrix with each entry on the diagonal being a small odd integer.

Claim 16. (original) The method of claim 12, wherein the at least one submatrix comprises an upper triangular matrix in which the diagonal entries are small odd integers.

Claim 17. (original) The method of claim 12, wherein the multiplying step comprises:

shifting a portion of the information word by a shift value corresponding to an entry of the parity check matrix;

performing a bitwise exclusive-OR of the shifted portion of the information word with a prior accumulated value;

storing the results of the step of the bitwise exclusive-OR as the accumulated value;

repeating the shifting, performing, and storing steps according to a weight value for the entry of the parity check matrix.

Claim 18. (original) The method of claim 17, wherein the multiplying step further comprises:

accumulating the last result of the repeated steps with selected stored right-hand side values.

Claim 19. (original) The method of claim 12, wherein the successively solving step comprises:

shifting a portion of selected contents of the right-hand side value memory by a shift value corresponding to an entry of the parity check matrix;

performing a bitwise exclusive-OR of the shifted portion of the contents of the right-hand side value memory with a prior accumulated value;

storing the results of the step of the bitwise exclusive-OR as the accumulated value; and

repeating the shifting, performing, and storing steps according to a weight value for the entry of the parity check matrix.

Claim 20. (original) The method of claim 12, wherein the step of determining corresponding parity bits from the stored right-hand side values comprises performing cyclically shifting selected stored right-hand side values.

Claim 21. (original) The method of claim 12, further comprising:

detecting a last submatrix as having row rank deficiency; and

responsive to the detecting step, solving a plurality of parity bits using an inverse of the last submatrix.

Claim 22. (original) The method of claim 12, further comprising:

detecting a last submatrix as having row rank deficiency and a weight of two; and

changing a parity bit position to a new information bit position;

then solving a plurality of parity bits corresponding to the last submatrix by successively applying a solved bit, beginning with a bit in the new information bit position, to equations corresponding to the last submatrix.

Claim 23. (original) The method of claim 12, wherein the updating step comprises:

shifting a portion of selected contents of the right-hand side value memory by a shift value corresponding to an entry of the parity check matrix;

performing a bitwise exclusive-OR of the shifted portion of the selected contents of the right-hand side value memory with a prior accumulated value;

storing the results of the step of the bitwise exclusive-OR as the accumulated value;

repeating the shifting, performing, and storing steps according to a weight value for the entry of the parity check matrix;

then accumulating the accumulated value with the selected contents of the right-hand side value memory.

Claim 24. (original) A transceiver for communicating encoded signals, comprising:

an encoder for encoding code words according to a low-density parity-check code, comprising:

a matrix value memory for storing shift and weight parameters associated with each of a plurality of macro matrix entries;

a register for storing at least a portion of an information bit vector; and

a cyclic multiply unit, coupled to the matrix value memory and to the register, and comprising:

a shifter for circularly shifting a selected portion of the information bit vector;

a first accumulator, for storing a result value; and

an exclusive-OR function, coupled to the shifter and to the accumulator, for performing a bitwise exclusive-OR operation on the output of the shifter and the output of the first accumulator; and

a right-hand side value memory, coupled to the cyclic multiply unit, for storing product values; and

a modulator, for modulating signals corresponding to the code words;

digital-to-analog converter circuitry, for generating an analog signal corresponding to the modulated signals; and

transmitter circuitry, for transmitting the analog signal over a communications facility.

Claim 25. (original) The transceiver of claim 24, wherein the encoder further comprises:

a parity memory, coupled to the cyclic multiply unit, for storing parity bits for the code word.

Claim 26. (original) The transceiver of claim 24, further comprising:

receiver circuitry, for receiving analog signals;

demodulator circuitry, for demodulating the received analog signals into code words; and

decoder circuitry, for decoding the code words to recover information words therefrom.

Claim 27. (original) The transceiver of claim 24, wherein the encoder further comprises:

an inverse submatrix memory, for storing parameters corresponding to an inverse of a submatrix; and

an inverse multiply unit, for multiplying product values by parameters stored in the inverse submatrix memory.

Claim 28. (original) The transceiver of claim 27, wherein the inverse submatrix memory is for storing a sequence of parameters for entries of the inverse of the submatrix;

and wherein the inverse multiply unit comprises:

a first register for storing a first portion of a product value;

a second register for storing a second portion of the product value;

a first logic circuit for combining the sequence of parameters with the first portion of the product value; and

a second logic circuit for combining the sequence of parameters, in a transposed sequence relative to that combined by the first logic circuit, with the second portion of the product value.

Claim 29. (original) The transceiver of claim 28, wherein the parameters stored in the inverse submatrix memory correspond to parameters along the diagonal and on one side of the diagonal of the submatrix.

Claim 30. (original) The transceiver of claim 24, wherein the encoder further comprises:

a parity memory, coupled to the cyclic multiply unit, for storing parity bits for the code word; and

a weight-two solution unit, having an input for receiving values from the right-hand side value memory, and having an output coupled to the parity memory, for successively solving parity bit values in a submatrix having a weight of two.

Claim 31. (original) The transceiver of claim 30, wherein the weight-two solution unit comprises:

a register for storing an indication of entries in the submatrix;

a multiplexer, coupled to the register, for selecting one of the bits of the register responsive to a control input;

a row index register, for storing a shifted value and for applying the value to the multiplexer as its control input;

logic circuitry, for generating an exclusive-OR of a prior result value and the bit selected by the multiplexer;

an output register, for storing results from the logic circuitry;

a demultiplexer, for coupling the logic circuitry to a position of the output register responsive to a control input; and

a column index register, for storing a shifted value and for applying the value to the demultiplexer as its control input.

Claims 32 & 33. (canceled)

Claim 34. (currently amended) A cyclic multiply unit, comprising:

a shifter for circularly shifting a selected portion of a vector;

a first accumulator, for storing a result value;

an exclusive-OR function, coupled to the shifter and to the accumulator, for performing a bitwise exclusive-OR operation on the output of the shifter and the output of the first accumulator; and

a second accumulator, having a first input for receiving a stored value, and a second input coupled to the output of the first accumulator, for updating the stored value with the contents of the first accumulator, ~~The cyclic multiply unit of claim 33,~~ wherein the second accumulator comprises a bitwise exclusive-OR function.